Private Multiplicative Weights

Jonathan Ullman, Northeastern University

Bar Ilan University Winter School: Differential Privacy. February 13, 2017.

Outline

- Privately releasing statistical queries via Private Multiplicative Weights
  - Originally: [Hardt-Rothblum'10], presentation is from [Gupta-Hardt-Roth-U'12, Hardt-Ligett-McSherry'13]
  - Optimal worst-case accuracy for statistical queries
  - Optimal worst-case computational efficiency
    - Although still exponential time
  - Extends to the case of online queries

Multiplicative Weights Update Rule

Day 1 investment	$\$.25$	$\$.25$	$\$.25$	$\$.25$
				
Day 1 losses	80%	60%	10%	40%

Day 1 total loss $.25 \times .8 \; + \; .25 \times .6 \; + \; .25 \times .1 \; + \; .25 \times .4$
$= \$0.475$

- You earn \$1 a day. You need a financier to manage it.

Multiplicative Weights Update Rule

Day 2 investment	$\$.10$	$\$.20$	$\$.50$	$\$.20$
				
Day 2 losses	10%	50%	80%	10%

Day 2 total loss $.10 \times .1 \; + \; .20 \times .5 \; + \; .50 \times .8 \; + \; .20 \times .1$
$= \$0.53$

- You earn \$1 a day. You need a financier to manage it.
- You decide to readjust your portfolio for day 2.

Multiplicative Weights Update Rule

Day 3 investment	$\$.20$	$\$.10$	$\$.20$	$\$.50$
				
Day 3 losses	20%	50%	50%	20%

Day 3 total loss $.20 \times .2 \; + \; .10 \times .5 \; + \; .20 \times .5 \; + \; .50 \times .2$
$= \$0.29$

- You earn \$1 a day. You need a financier to manage it.
- You decide to readjust your portfolio for day 2.
- And so on and so forth…

Multiplicative Weights Update Rule

Day t investment	p_1^t	p_2^t	…	p_m^t
				
Day t losses	ℓ_1^t	ℓ_2^t	…	ℓ_m^t
Day t total loss	$p_1^t \ell_1^t \quad +$	$p_2^t \ell_2^t \quad +$	$\ldots \quad +$	$p_m^t \ell_m^t = \langle p^t, \ell^t \rangle$

- T days, m actions you can take each day, losses ℓ_i^t
- Total loss is $\sum_{t=1}^{T} \langle p^t, \ell^t \rangle$
- Playing i every day, you would have lost $\min_i \sum_{t=1}^{T} \ell_i^t$
- $R_T(\ell) = \sum_{t=1}^{T} \langle p^t, \ell^t \rangle - \min_i \sum_{t=1}^{T} \ell_i^t$
  - Regret: How dumb do I feel for not sticking with i?

Multiplicative Weights Update Rule

Day t investment	p_1^t	p_2^t	…	p_m^t
				
Day t losses	ℓ_1^t	ℓ_2^t	…	ℓ_m^t
Day t total loss	$p_1^t \ell_1^t \quad +$	$p_2^t \ell_2^t \quad +$	$\ldots \quad +$	$p_m^t \ell_m^t = \langle p^t, \ell^t \rangle$

- Def [Regret]: $R_T(\ell) = \sum_{t=1}^{T} \langle p^t, \ell^t \rangle$ - $\min_i \sum_{t=1}^{T} \ell_i^t$
- Theorem [Littlestone-Warmuth'94]:
  There is an algorithm, MWU, that guarantees regret $R_T(\ell) \leq 2\sqrt{T \ln(m)}$ for any sequence of losses $\{\ell_i^t\}$.

Multiplicative Weights Update Rule

Start with equal weights

Let $\boldsymbol{\eta} = \mathbf{min}\{\sqrt{\mathbf{ln}(\boldsymbol{m})/\boldsymbol{T}}, \mathbf{1}\}$, $\boldsymbol{w}^{\mathbf{1}} = \vec{\mathbf{1}}$, $\boldsymbol{p}^{\mathbf{1}} = \boldsymbol{w}^{\mathbf{1}}/\boldsymbol{m}$,

For $\boldsymbol{t} = \mathbf{1}, \dots, \boldsymbol{T}$:

Receive losses $\boldsymbol{\ell}^{\boldsymbol{t}} = (\boldsymbol{\ell}_{\mathbf{1}}^{\boldsymbol{t}}, \dots, \boldsymbol{\ell}_{\boldsymbol{m}}^{\boldsymbol{t}})$

Set $\boldsymbol{w}_{\boldsymbol{i}}^{\boldsymbol{t}+\mathbf{1}} = \mathbf{exp}(-\boldsymbol{\eta}\boldsymbol{\ell}_{\boldsymbol{i}}^{\boldsymbol{t}}) \cdot \boldsymbol{w}_{\boldsymbol{i}}^{\boldsymbol{t}}$ for every $\boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{m}$

Set $\boldsymbol{p}_{\boldsymbol{i}}^{\boldsymbol{t}+\mathbf{1}} = \left(\boldsymbol{w}_{\boldsymbol{i}}^{\boldsymbol{t}+\mathbf{1}}\right)/\left(\sum_{\boldsymbol{j}} \boldsymbol{w}_{\boldsymbol{j}}^{\boldsymbol{t}+\mathbf{1}}\right)$

Large loss makes prob. go down
Small loss makes prob. go up

- Def [Regret]: $\boldsymbol{R}_{\boldsymbol{T}}(\boldsymbol{\ell}) = \sum_{\boldsymbol{t}=\mathbf{1}}^{\boldsymbol{T}} \langle \boldsymbol{p}^{\boldsymbol{t}}, \boldsymbol{\ell}^{\boldsymbol{t}} \rangle - \min_{\boldsymbol{i}} \sum_{\boldsymbol{t}=\mathbf{1}}^{\boldsymbol{T}} \boldsymbol{\ell}_{\boldsymbol{i}}^{\boldsymbol{t}}$

- Theorem [Littlestone-Warmuth'94]:
  There is an algorithm, MWU, that guarantees regret $R_T(\ell) \leq 2\sqrt{T \ln(m)}$ for any sequence of losses $\{\ell_i^t\}$.

Multiplicative Weights Update Rule

Let $\eta = \min\{\sqrt{\ln(m)/T}, 1\}$, $w^1 = \vec{1}$, $p^1 = w^1/m$,
For $t = 1, \ldots, T$:
 Receive losses $\ell^t = (\ell_1^t, \ldots, \ell_m^t)$
 Set $w_i^{t+1} = \exp(-\eta \ell_i^t) \cdot w_i^t$ for every $i = 1, \ldots, m$
 Set $p_i^{t+1} = (w_i^{t+1})/(\sum_j w_j^{t+1})$

- Idea: Use sum of weights $W^t = \sum_j w_j^t$ as a potential fn.
  - $W^1 = m$
  - $\exp(-\eta \sum_t \ell_i^t) \leq W^T$
    - (loss of i lower bounds W^T)
  - $W^T \leq m \cdot exp(\eta^2 T - \eta \sum_t \langle p^t, \ell^t \rangle)$
    - (your loss upper bounds W^T)
  - Algebra $\Longrightarrow \sum_t \langle p^t, \ell^t \rangle - \sum_t \ell_i^t \leq \eta T + \frac{\ln(m)}{\eta}$

Multiplicative Weights Update Rule

Let $\boldsymbol{\eta} = \mathbf{min}\{\sqrt{\mathbf{ln}(\boldsymbol{m})/\boldsymbol{T}}, \mathbf{1}\}$, $\boldsymbol{w}^{\mathbf{1}} = \vec{\mathbf{1}}$, $\boldsymbol{p}^{\mathbf{1}} = \boldsymbol{w}^{\mathbf{1}}/\boldsymbol{m}$,
For $\boldsymbol{t} = \mathbf{1}, \dots, \boldsymbol{T}$:
Receive losses $\ell^{t} = (\ell_{1}^{t}, \dots, \ell_{m}^{t})$
Set $\boldsymbol{w}_{i}^{t+1} = \mathbf{exp}(-\boldsymbol{\eta}\ell_{i}^{t}) \cdot \boldsymbol{w}_{i}^{t}$ for every $\boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{m}$
Set $\boldsymbol{p}_{i}^{t+1} = \left(\boldsymbol{w}_{i}^{t+1}\right)/\left(\sum_{j} \boldsymbol{w}_{j}^{t+1}\right)$

- Def [Regret]: $\boldsymbol{R_T}(\ell) = \sum_{t=1}^{T}\langle \boldsymbol{p}^{t}, \ell^{t}\rangle - \min_{i} \sum_{t=1}^{T} \ell_{i}^{t}$
- Theorem [Littlestone-Warmuth'94]:
  There is an algorithm, MWU, that guarantees regret
  $R_T(\ell) \leq 2\sqrt{T \ln(m)}$ for any sequence of losses $\{\ell_i^t\}$.

Multiplicative Weights Update Rule

Let $\boldsymbol{\eta} = \mathbf{min}\{\sqrt{\mathbf{ln}(\boldsymbol{m})/\boldsymbol{T}}, \mathbf{1}\}$, $\boldsymbol{w}^{\mathbf{1}} = \vec{\mathbf{1}}$, $\boldsymbol{p}^{\mathbf{1}} = \boldsymbol{w}^{\mathbf{1}}/\boldsymbol{m}$,
For $\boldsymbol{t} = \mathbf{1}, \dots, \boldsymbol{T}$:
Receive losses $\ell^{\boldsymbol{t}} = (\ell_{\mathbf{1}}^{\boldsymbol{t}}, \dots, \ell_{\boldsymbol{m}}^{\boldsymbol{t}})$
Set $\boldsymbol{w}_{\boldsymbol{i}}^{\boldsymbol{t}+\mathbf{1}} = \mathbf{exp}(-\boldsymbol{\eta}\ell_{\boldsymbol{i}}^{\boldsymbol{t}}) \cdot \boldsymbol{w}_{\boldsymbol{i}}^{\boldsymbol{t}}$ for every $\boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{m}$
Set $\boldsymbol{p}_{\boldsymbol{i}}^{\boldsymbol{t}+\mathbf{1}} = \left(\boldsymbol{w}_{\boldsymbol{i}}^{\boldsymbol{t}+\mathbf{1}}\right)/\left(\sum_{\boldsymbol{j}} \boldsymbol{w}_{\boldsymbol{j}}^{\boldsymbol{t}+\mathbf{1}}\right)$

- Def [Regret]: $\boldsymbol{R}_{\boldsymbol{T}}(\ell) = \sum_{\boldsymbol{t}=\mathbf{1}}^{\boldsymbol{T}} \langle \boldsymbol{p}^{\boldsymbol{t}}, \ell^{\boldsymbol{t}} \rangle - \min_{\boldsymbol{p}^*} \sum_{\boldsymbol{t}=\mathbf{1}}^{\boldsymbol{T}} \langle \boldsymbol{p}^*, \ell^{\boldsymbol{t}} \rangle$
- Theorem [Littlestone-Warmuth'94]:
There is an algorithm, MWU, that guarantees regret
$R_T(\ell) \leq 2\sqrt{T \ln(m)}$ for any sequence of losses $\{\ell_i^t\}$.

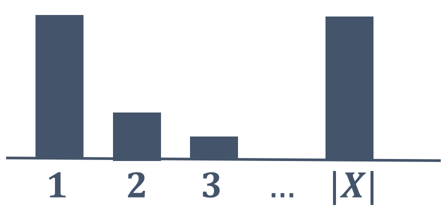
Outline

- Privately releasing statistical queries via Private Multiplicative Weights
  - Originally: [Hardt-Rothblum'10], presentation is from [Gupta-Hardt-Roth-U'12, Hardt-Ligett-McSherry'13]
  - Optimal worst-case accuracy for statistical queries
  - Optimal worst-case computational efficiency
    - Although still exponential time
  - Extends to the case of online queries

Towards PMW

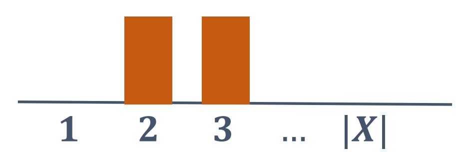
- Dataset x is a probability distribution p^* over X
- Want to answer queries $\langle q, p^* \rangle$ for $q \in Q$
  - Statistical query $\frac{1}{n}\sum_i \phi(x_i)$ becomes $\langle q, p^* \rangle$ where $q = (\phi(1), \phi(2), \dots)$
- Error of $\hat{p}$ is $\max_{q \in Q} \langle q, \hat{p} - p^* \rangle$

Why only one sided?



p^*

Think of $x \in X^n$ as a distribution over X



q^1

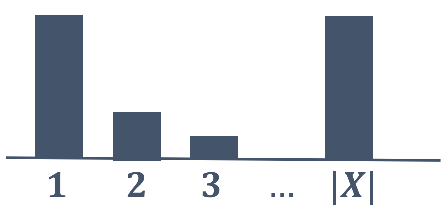
q is a vector over X

Towards PMW

$p^1 = \text{Uniform}(X)$
For $t = 1, \dots, T$:
  Find $q^t = \underset{q \in Q}{\text{argmax}} \langle q, p^t - p^* \rangle$
  Let $p^{t+1} = MWU(p^t, q^t)$
Output $\hat{p} = \frac{1}{T} \sum_t p^t$

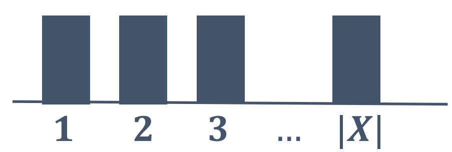
Find a "bad query" q^t

Multiplicative weight update using losses $\ell^t = q^t$



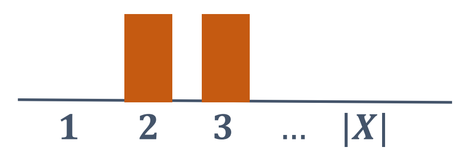
p^*

Think of $x \in X^n$ as a distribution over X



p^1

Approximation p^t is also a distribution over X



q^1

q^t is a vector over X where p^*, p^t are very different

Towards PMW

$p^1 = \mathrm{Uniform}(X)$

For $t = 1, \dots, T$:

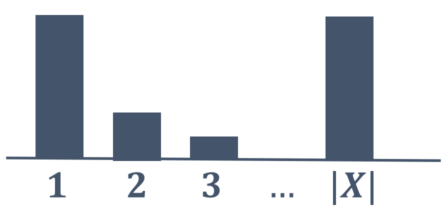
  Find $q^t = \underset{q \in Q}{\mathrm{argmax}} \langle q, p^t - p^* \rangle$

  Let $p^{t+1} = MWU(p^t, q^t)$

Output $\hat{p} = \frac{1}{T} \sum_t p^t$

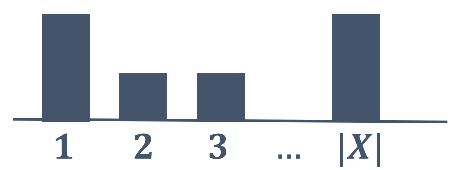
Find a "bad query" q^t

Multiplicative weight update using losses $\ell^t = q^t$



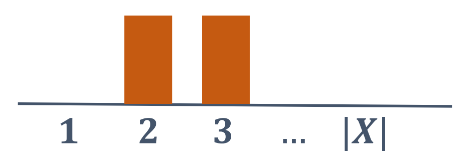
p^*

Think of $x \in X^n$ as a distribution over X



p^2

Use q^t to make p^{t+1} closer to p^*



q^1

q^t is a vector over X where p^*, p^t are very different

Towards PMW

$p^1 = \text{Uniform}(X)$
For $t = 1, \ldots, T$:
  Find $q^t = \underset{q \in Q}{\text{argmax}} \langle q, p^t - p^* \rangle$
  Let $p^{t+1} = MWU(p^t, q^t)$
Output $\hat{p} = \frac{1}{T} \sum_t p^t$

Find a "bad query" q^t

Multiplicative weight update using losses $\ell^t = q^t$

Claim: For every $q \in Q$, $\hat{p}$ has error at most $2\sqrt{\frac{\ln|X|}{T}}$

$$\max_{q \in Q} q \cdot \left(\frac{1}{T} \sum_t p^t - p^* \right) \leq \frac{1}{T} \sum_t \max_{q \in Q} \langle q, p^t - p^* \rangle$$
$$= \frac{1}{T} \sum_t \langle q^t, p^t - p^* \rangle$$
$$\leq \frac{1}{T} R_T \leq 2\sqrt{\frac{\ln|X|}{T}}$$

Putting the Private in Private MW

$p^1 = \mathrm{Uniform}(X)$

For $t = 1, \ldots, T$:

Find $q^t = \underset{q \in Q}{\mathrm{argmax}} \langle q, p^t - p^* \rangle$

This is the only step where we use the dataset p^*

Let $p^{t+1} = MWU(p^t, q^t)$

Output $\hat{p} = \frac{1}{T} \sum_t p^t$

Q: How can we find the (approximately) worst query privately?

A: The exponential mechanism or find-noisy-max!

Exponential Mechanism

$EM_{\varepsilon_0,Q}(p^t - p^*)$:

Choose q^t with probability proportional to

$$\exp\left(\frac{\varepsilon_0}{2} \cdot n \cdot \langle q^t, p^t - p^* \rangle\right)$$

Output q^t

- Recall: $EM_{\varepsilon_0,Q}$ is $(\varepsilon_0, 0)$-dp
  - To achieve (ε, δ)-dp over T iterations, set $\varepsilon_0 \approx \frac{\varepsilon}{\sqrt{T \ln(1/\delta)}}$
- Recall: $\mathbb{E}[\langle q^t, p^t - p^* \rangle] \geq \max_{q \in Q} \langle q, p^t - p^* \rangle \; - \frac{2 \ln|Q|}{\varepsilon_0 n}$

Call this α_0

Putting the Private in Private MW

$p^1 = \mathrm{Uniform}(X)$
For $t = 1, \dots, T$:
    Privately sample $q^t \leftarrow EM_{\varepsilon_0, Q}(p^t - p^*)$
    Let $p^{t+1} = MWU(p^t, q^t)$
Output $\hat{p} = \frac{1}{T}\sum_t p^t$

The whole algorithm is now private.

Claim: For every $q \in Q$, $\hat{p}$ has error at most $2\sqrt{\frac{\ln|X|}{T}} + \frac{2\ln|Q|}{\varepsilon_0 n}$

$$
\begin{aligned}
\max_{q \in Q} q \cdot \left(\frac{1}{T} \sum_t p^t - p^* \right) &\leq \frac{1}{T} \sum_t \max_{q \in Q} \langle q, p^t - p^* \rangle \\
&= \frac{1}{T} \textstyle\sum_t \langle q^t, p^t - p^* \rangle + \alpha_0 \\
&\leq \frac{1}{T} R_T + \alpha_0 \leq 2\sqrt{\frac{\ln|X|}{T}} + \alpha_0
\end{aligned}
$$

Putting the Private in Private MW

$p^1 = \text{Uniform}(X)$
For $t = 1, \dots, T$:
  Privately sample $q^t \leftarrow EM_{\varepsilon_0,Q}(p^t - p^*)$
  Let $p^{t+1} = MWU(p^t, q^t)$
Output $\hat{p} = \frac{1}{T}\sum_t p^t$

The whole algorithm is now private.

Claim: For every $q \in Q$, $\hat{p}$ has error at most $2\sqrt{\frac{\ln|X|}{T}} + \frac{2\ln|Q|}{\varepsilon_0 n}$

To achieve (ε, δ)-dp, set $\varepsilon_0 = \varepsilon/\sqrt{T\ln(1/\delta)}$

Putting the Private in Private MW

$p^1 = \text{Uniform}(X)$
For $t = 1, \dots, T$:
  Privately sample $q^t \leftarrow EM_{\varepsilon_0,Q}(p^t - p^*)$
  Let $p^{t+1} = MWU(p^t, q^t)$
Output $\hat{p} = \frac{1}{T}\sum_t p^t$

The whole algorithm is now private.

Claim: For every $q \in Q$, $\hat{p}$ has error $2\sqrt{\frac{\ln|X|}{T}} + \frac{2\ln|Q|\sqrt{T\ln(1/\delta)}}{\varepsilon n}$

To achieve (ε, δ)-dp, set $\varepsilon_0 = \varepsilon/\sqrt{T\ln(1/\delta)}$

Now, set T optimally

Putting the Private in Private MW

$p^1 = \mathrm{Uniform}(X)$
For $t = 1, \ldots, T$:
  Privately sample $q^t \leftarrow EM_{\varepsilon_0, Q}(p^t - p^*)$
  Let $p^{t+1} = MWU(p^t, q^t)$
Output $\hat{p} = \frac{1}{T}\sum_t p^t$

The whole algorithm is now private.

Claim: For every $q \in Q$, $\hat{p}$ has error $\alpha = O\left(\frac{\ln|Q| \sqrt{\ln|X| \cdot \ln(1/\delta)}}{\varepsilon n}\right)^{1/2}$

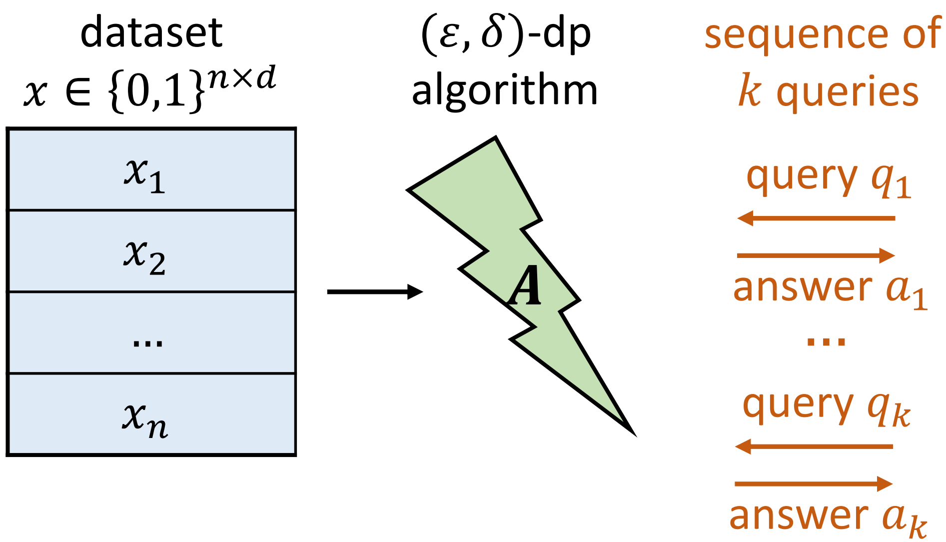
Running time is $O(T|Q||X|)$. Each round is essentially linear in the size of the set of queries Q, which can be an arbitrary $|Q| \times |X|$ matrix. But can be exponential in the size of the dataset.

Outline

- Privately releasing statistical queries via Private Multiplicative Weights [Hardt-Rothblum'10].
  - Optimal worst-case accuracy for statistical queries
  - Optimal worst-case computational efficiency
    - Although still exponential time
  - Extends to the case of online queries

Extension to Online Queries

- Queries $q_1, \ldots, q_k$ arrive one at a time (can be adaptive)
- Must give an α-accurate answer a_t after each query q_t



answers accurate if

$$\max_j \left| q_j(x) - a_j \right| \le \alpha$$

Online MW

$p^1 = \text{Uniform}(X), t = 1, T \approx \frac{4 \ln |X|}{\alpha^2}, \varepsilon_0 \approx \frac{\varepsilon}{\sqrt{T \ln(1/\delta)}}$

Repeat until $t = T$: [outer loop]

Repeat: [inner loop]

Let q be the next query

If $|\langle q, p^t - p^* \rangle| \leq \alpha$, output $a = \langle q, p^t \rangle$

Else output $a = \langle q, p^* \rangle + Lap\left(\frac{1}{\varepsilon_0 n}\right)$, break loop, and UPDATE

UPDATE:

Define ℓ^t to be either q or $-q$ depending on the sign of the error

Let $p^{t+1} = MWU(p^t, \ell^t)$, let $t = t + 1$

If p^t is accurate, use it to answer

Otherwise use Laplace

Key Claim: There are only $T = \Theta\left(\frac{\ln|X|}{\alpha^2}\right)$ UPDATES

Online Private MW

$p^1 = \text{Uniform}(X), t = 1, T \approx \frac{4 \ln |X|}{\alpha^2}, \varepsilon_0 \approx \frac{\varepsilon}{\sqrt{T \ln(1/\delta)}}$

Repeat until $t = T$: [outer loop]

Let $\hat{\alpha} = \alpha + Lap\left(\frac{1}{\varepsilon_0 n}\right)$

Repeat: [inner loop]

Let q be the next query

If $|\langle q, p^t - p^* \rangle| + Lap\left(\frac{1}{\varepsilon_0 n}\right) \leq \hat{\alpha}$, output $a = \langle q, p^t \rangle$

Else output $a = \langle q, p^* \rangle + Lap\left(\frac{1}{\varepsilon_0 n}\right)$, break loop, and UPDATE

UPDATE:

Define ℓ^t to be either q or $-q$ depending on the sign of the error

Let $p^{t+1} = MWU(p^t, \ell^t)$, let $t = t + 1$

Randomize the threshold

Randomize the tests

Key Claim: This algorithm is the composition of $T = \Theta\left(\frac{\ln|X|}{\alpha^2}\right)$ instances of the sparse vector primitive.

Master Theorem for Query Release

- Theorem [Hardt-Rothblum'10]: We can privately answer any sequence of k online (and adaptively chosen) queries in time $O(|X|+|Q|)$ per query with error at most

$$\alpha = O\left(\frac{\ln|Q|\,\sqrt{\ln|X|\cdot\ln(1/\delta)}}{\varepsilon n}\right)^{1/2}$$

Outline

- Privately releasing statistical queries via Private Multiplicative Weights [Hardt-Rothblum'10].
  - Optimal worst-case accuracy for statistical queries
  - Optimal worst-case computational efficiency
    - Although still exponential time
  - Extends to the case of online queries